

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Gregorič

**Spletna aplikacija za vnos in izpis napak  
na proizvodnih linijah**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2014

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

## **Tematika dela**

V okviru diplomske naloge razvijte spletno aplikacijo za FTQ (angl. first time quality) sestanke. Aplikacija naj omogoča čim enostavnejši uporabniški vmesnik za vnos in izpis napak na proizvodnih linijah. Uporabniški vmesnik naj se čim manj razlikuje od zdajšnjega pripravljenega v Excel-u. Aplikacija naj omogoča več vrst izpisa, med njimi naj bo tudi izpis napak z dvema grafoma za oglasno desko, ki naj bo natanko enak zdajšnjemu prav tako pripravljenemu v programu Excel. Aplikacija naj bo dostopna na Intranetu, ki je v lasti podjetja. Ob prijavi naj prepozna uporabnika, njegovo pripadnost liniji in omogoči vnos ali izpis napak samo za njegovo linijo.

# **IZJAVA O AVTORSTVU**

## **diplomskega dela**

Spodaj podpisani      Boštjan Gregorič,  
z vpisno številko      63100027,

sem avtor diplomskega dela z naslovom:

**Spletna aplikacija za vnos in izpis napak na proizvodnih linijah**

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luka Šajna;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela;
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 24. 4. 2014

Podpis avtorja:

## **Zahvala**

Zahvaljujem se mentorju doc. dr. Luki Šajn za pomoč in koristne nasvete pri izdelavi diplomskega dela.

Posebna zahvala gre sodelavcu Aleksandru Šinigoj za vse nasvete in pomoč pri razvoju spletne aplikacije.

Zahvaljujem se tudi moji družini, ki me je spodbujala in stala ob strani vsa leta študija.

# Kazalo vsebine

Zahvala

Seznam uporabljenih kratic in simbolov

Povzetek

Abstract

1.	Uvod .....	1
2.	O aplikaciji .....	2
2.1	Predstavitev podjetja .....	2
2.2	Namen aplikacije .....	3
3.	Razvojna orodja in metode .....	4
4.	Uporabljena programska orodja .....	4
4.1	Eclipse IDE for Java EE Developers .....	4
4.2	Spletni strežnik Tomcat Apache .....	4
4.3	IML .....	5
4.4	Oracle SQL Developer .....	7
5.	Uporabljene metode .....	8
5.1	XML .....	8
5.2	HTML .....	8
5.3	CSS .....	8
5.4	JavaScript .....	9
5.5	SQL .....	9
5.6	SVN .....	9
5.6.1	SVN – direktoriji .....	10
5.6.2	SVN - merge urejanje konfliktov .....	10
6.	Razvoj aplikacije .....	11
6.1	Arhitektura sistema .....	11
6.2	Podatkovna baza .....	12
6.3	Uporabniški vmesnik .....	12
6.3.1	Vnos napak .....	13
6.3.1.1	Vnos napak na star način .....	13

6.3.1.2	Vnos napak z aplikacijo .....	15
6.3.2	Izpis napak.....	18
6.3.3	Mesečni izpis napak.....	20
6.3.4	Tedenski izpis napak z dvema grafoma.....	21
6.3.5	Šifrant napak.....	22
7.	Sklepne ugotovitve .....	24
	Literatura .....	25
	Seznam slik.....	27
	Priloge.....	I
	Priloga 1.....	I
	Priloga 2.....	II
	Priloga 3.....	IV

## Seznam uporabljenih kratic in simbolov

PPM	– procent na milijon (angl. percent per milion)
ODS	– osnovna delovna skupina
FTQ	– (angl. First Time Quality)
SAP	– ime informacijskega sistema
HTML	– označevalni jezik za izdelavo spletnih strani (angl. Hyper Text Markup Language)
CSS	– stilne predloge (angl. Cascading Style Sheets)
JavaScript	– objektni skriptni programski jezik za razvoj interaktivnih spletnih strani
SQL	– strukturirani povpraševalni jezik za delo s podatkovnimi bazami (angl. Structured Query Language)
XML	– razširljiv označevalni jezik (angl. Extensible Markup Language)
IML	– razširljiv označevalni jezik podjetja Letrika d.d.
SVN	– sistem za verzioniranje dokumentov (angl. Apache™ Subversion®)



## **Povzetek**

V diplomski nalogi je predstavljena spletna aplikacija za vnos in izpis napak na proizvodnih linijah, ter njen razvoj. Namenjena je delavcem v podjetju Letrika d.d. za beleženje napak ter analiziranje le teh po različnih linijah.

V prvem delu naloge je predstavljen dosednji postopek zajema in analize napak, kateri je potekal ročno brez spletne aplikacije. Predstavljena sta tudi cilj in namen aplikacije.

Glavni del diplomskega dela opisuje postopek izdelave aplikacije. Njen uporabniški vidik predstavlja funkcionalnosti, njeno uporabo ter učinkovitost. Predstavljen je tudi tehnični vidik, vse tehnologije in programska orodja, katere so bile v aplikaciji uporabljene ter programski jezik v katerem je napisana.

Zadnji del vsebuje zaključek, ter spoznanja s katerimi smo se soočili tekom izdelave. Presojajo ali aplikacija vsebuje vse funkcionalnosti, katere so bile izražene s strani firme. Predstavljene so tudi možnosti nadgradnje in izboljšave aplikacije.

**Ključne besede:** spletna aplikacija, PPM, FTQ, podatkovna baza, IML, XML, vnos napak.

## **Abstract**

The diploma thesis deals with web application for entry and display of errors in the production lines and its development. It is designed for the workers of the company Letrika d.d. to record errors and analyze them by different lines.

The first part presents the recent process of capture and analysis of errors that was carried out manually and with no web application. It also gives the goal and the purpose of the application.

The main part of the thesis describes the making of the application. Its user point of view shows performance, use, and efficiency. In addition, a technical view is given showing all technologies and software tools used in the application and the programming language, in which it was written.

The last part concludes with the findings that were reached during the making and an assessment whether the application contains all functionalities that were expressed and required by the company. Moreover, possibilities of the application upgrade and enhancement are described.

**Key words:** web application, PPM, FTQ, database, IML, XML, entry errors.

# 1. Uvod

Danes si brez računalnikov beleženje podatkov sploh ne moremo predstavljati, saj je uporaba računalnikov iz dneva v dan večja. Nekoč so bili vsi podatki natisnjeni na papirju, ter shranjeni v arhivih. Vsakodnevno pa težimo k temu, da bi vse te natisnjene podatke prenesli iz arhiva v računalniške sisteme. Taki sistemi omogočajo hitrejše delo s podatki ki jih hranimo, ter preprečujejo nastajanje napak pri uporabi le teh. Nad podatki veliko lažje izvajamo različne analize tudi pri velikem številu le teh. Vsi sistemi so dan danes povezani v različna omrežja, po katerih komunicirajo med seboj. Tako nam omogočajo, da iz različnih lokacij dostopamo do istih podatkov, ki so lahko shranjeni na drugi lokaciji.

Kot vsaka stvar ima tudi beleženje podatkov v računalniških sistemih svojo slabo stran. Vsak računalniški sistem je povezan v svetovno medmrežje, saj drugače ni mogoče dostopati iz različnih lokacij do njega. S povezavo na svetovni splet lahko postanemo tarča nepridipravov. Sistemi kateri hranijo pomembne podatke, morajo imeti različne varnostne prepreke, da jih zaščitijo pred dostopom nepridipravov. Te osebe lahko z vdorom v bazo podatkov uničijo vse podatke ali pa jih uporabijo za zlorabo.

Z namenom lažjega beleženja podatkov, ter hitrejšega manipuliranja z njimi je bila izdelana spletna aplikacija, katera je podrobneje opisana v glavnem delu. Namenjena je podjetju Letrika d.d. za beleženje in analizo napak. Do sedaj je beleženje potekalo najprej na papirju in kasneje v elektronskih preglednicah (Excel).

Pred izdelavo spletne aplikacije, so bila izdana navodila, ter želene funkcionalnosti za izdelavo s strani podjetja.

Aplikacija vsebuje naslednje (podprograme) funkcionalnosti:

- Vnos napak
  - Vnosno formo, preko katere se bodo napake vnašale in shranjevale v podatkovno bazo
- Mesečni pregled napak
  - Izpis napak, ki so se pojavile v enem mesecu
- Pregled napak za željeno obdobje
- Analiziran izpis petih napak, ki so se pojavile v enem tednu
  - Izpis je posebej prilagojen za informativno tablo
  - Dodana tudi dva grafa
- Ažuriranje šifranta napak
  - Dodajanje, brisanje ali spreminjanje napak na vnosni formi

## **2. O aplikaciji**

Aplikacija je izdelana za podjetje Letrika d.d. in je namenjena vnosu in analiziranju napak po linijah. Na posamezni liniji izdelujejo različne izdelke, kjer se pojavljajo raznorazne napake. Shranjenim napakam se bo dodelilo ukrep s katerim, naj bi se jih odpravilo.

### **2.1 Predstavitev podjetja**

Letrika je globalna skupina podjetij, s sedežem matične družbe v Sloveniji, ki svoje kakovostne, inovativne in k sodobnim trendom usmerjene izdelke in storitve trži po skoraj vsem svetu.

Z več kot 2600 zaposlenimi v skupini so usmerjeni v proizvodnjo, razvoj in prodajo zaganjalnikov, alternatorjev, električnih pogonskih in mehatronskih sistemov, vse bolj pa se usmerjajo tudi k zelenim tehnologijam in ponujajo tudi izdelke s področja izrabe obnovljivih virov energije ali zmanjševanja škodljivih vplivov na okolje.

Njihovi kupci so največji globalni proizvajalci kmetijske in gradbene mehanizacije, različnih delovnih strojev in široke palete transportnih sredstev, vključno z avtomobilsko industrijo ter industrijo plovil in golf vozičkov. Med najpomembnejše trge prištevajo trge zahodne Evrope in ZDA, vse pomembnejši pa postajajo tudi trgi v razvoju in države BRIK.

Več kot 50 let razvoja, izkušenj ter strokovnih dosežkov in uspehov so leta 2012 združili v novo blagovno znamko Letrika, pod katero bodo v prihodnje še dodatno krepili vezi z vsemi partnerji, odjemalci in ostalimi deležniki.[6]

## 2.2 Namen aplikacije

Namen aplikacije je nadomestiti sedanji način vnosa in beleženja napak v elektronskih preglednicah z aplikacijo, ki bo podatke hranila v podatkovni bazi. S tem bo lažje dostopati do napak, jim dodeliti ukrep in jih odpraviti.

Beleženje napak v proizvodnji je ključnega pomena, saj bi bilo poslovanje podjetja z velikim številom napak zelo slabo. Da bi te napake odpravili, si jih moramo zabeležiti in kasneje pravilno ukrepati, da se v nadaljevanju ne bi več ponovile.

Star način beleženja napak je bil zamuden in pri tem je lahko prišlo tudi do napak pri vnosu podatkov. Z dneva v dan so elektronske preglednice postajale vse večje. Za vsaki teden se je naredila nova elektronska preglednica. Analize za nazaj so bile težko izvedljive, pogled nad podatki je bil manj pregleden. S tem namenom je bila izdelana spletna aplikacija, da bi beležila napake, ki se pojavijo po linijah.

Po novem se bodo podatki o napakah hranili vsi na enem mestu. Analiza nad njimi ne bo več težava, ker bo dostop do podatkov v podatkovni bazi zelo hiter in enostaven.

Za dostop do podatkov v podatkovni bazi ima aplikacija funkcijo, ki omogoča vodjem in tehnologom vpogled nad zabeleženimi podatki za neko obdobje ali celo en mesec.

Pravilnik predpisuje, da morajo na informativni tabli pred vsako linijo biti natisnjene določene analize. Ena izmed njih je prikaz petih napak, katere so se pojavile prejšnji teden z izračunanim PPM-om (angl. percent per milion) in grafom seštetih napak za pol leta nazaj. V ta namen vsebuje aplikacija funkcijo za izpis napak in izris grafa.

Vsaka linija ima svoj šifrant napak, ki se pojavljajo. Ta šifrant je zabeležen prav tako v bazi podatkov. Včasih se lahko pojavi tudi kakšna nova napaka, zato jo je potrebno zabeležiti v bazo. Aplikacija je tako zasnovana, da nam omogoča vnos, spremembo in brisanje napak.

### **3. Razvojna orodja in metode**

Za razvoj aplikacij obstaja veliko orodij. Vsak si izbere tako orodje, da pri programiranju nima težav z uporabo. V nadaljevanju so opisana programska orodja in metode, ki so bila uporabljena pri razvoju aplikacije.

## **4. Uporabljena programska orodja**

### **4.1 Eclipse IDE for Java EE Developers**

Aplikacija je napisana v programskem orodju Eclipse developer tools. Eclipse je večjezično razvojno okolje, ki vključuje osnovno delovno okolje in razširljivi sistem (plug-in system) za prilagajanje okolja za druge platforme. Napisan je v programskem jeziku Java, zato se ga uporablja za pisanje aplikacij v Javi, ampak s pomočjo različnih vtičnikov lahko uporabljamo tudi druge programske jezike kot so HTML, C, C++, PHP...

Delo v Eclipse-u je razdeljeno na enega ali več tako imenovanih delovnih okolji (angl. Workspace). To je mapa na lokalnem disku, kjer so shranjeni programi. V okviru enega delovnega okolja imamo posamezne projekte.[\[7\]](#)

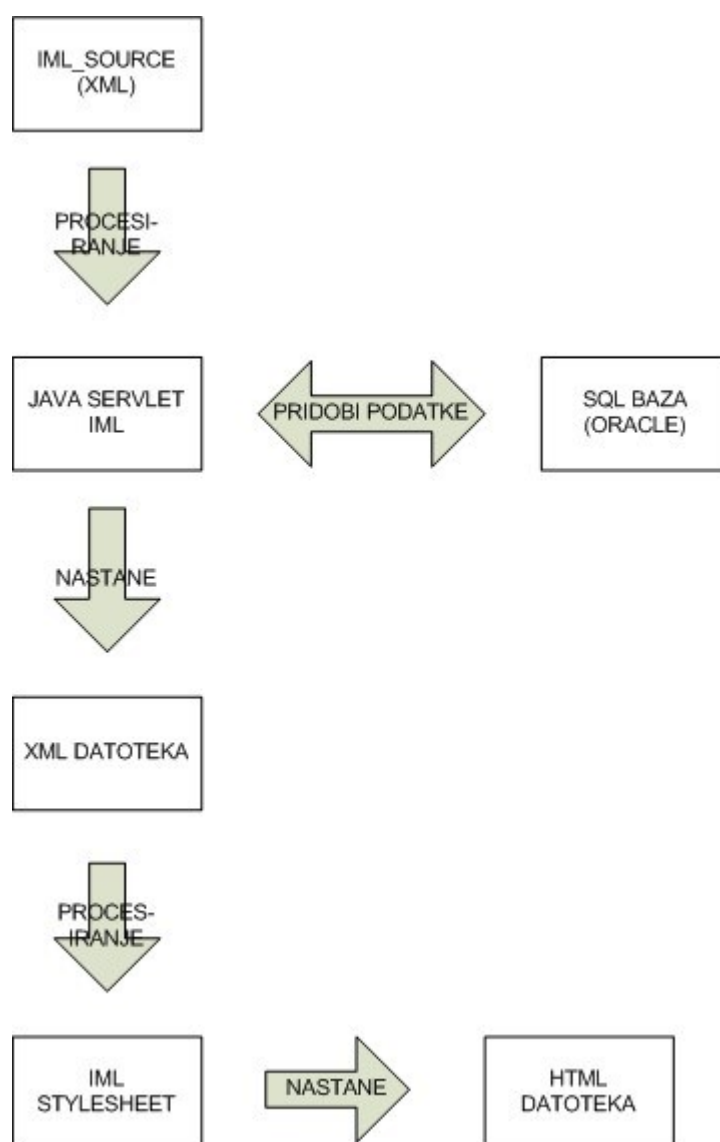
### **4.2 Spletni strežnik Tomcat Apache**

Je spletni strežnik, ki igra ključno vlogo pri širjenju spleta. Bil je prva alternativa Netscapeovemu spletnemu strežniku, trenutno znanemu kot spletni strežnik Sun Java System. Od aprila 1996 je Apache najbolj popularen http strežnik na celotnem spletu od oktobra 2007 pa je bilo na Apachijevem strežniku postavljenih približno 48% vseh spletnih strani.

Spletni vmesnik omogoča upravljanje s spletnim strežnikom Apache, ki je nameščen na virtualnem strežniku. Apache lahko poganjamo tudi direktno iz Eclipse okolja. S tem dobimo možnost preizkusa dela na lastnem računalniku in to brez tega, da bi motili druge uporabnike.[\[2\]](#)

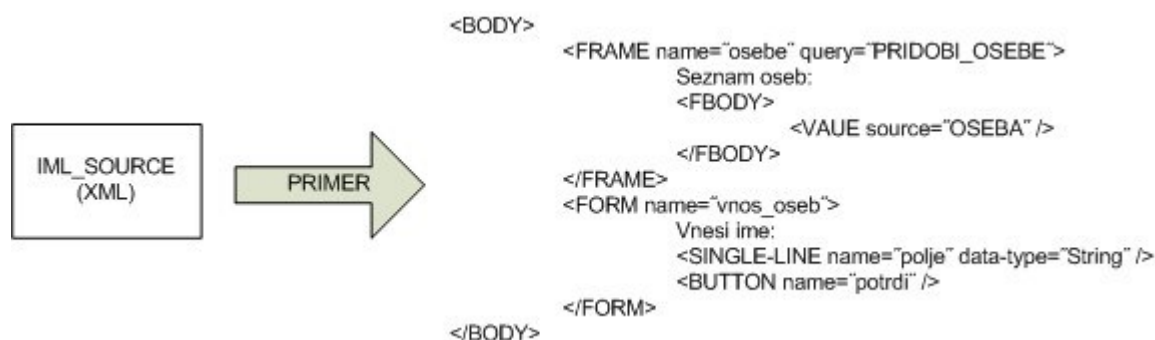
### 4.3 IML

Podjetje si je za lažji razvoj spletnih aplikacij razvilo svojo rešitev, ki temelji na XML tehnologiji. Programi (strani) so kodirani v XML dokumentih (IML programi). Program se izvaja preko Java Servlet-a, ki deluje na Apache TomCat strežniku. Za izvajanje so shranjeni v podatkovni bazi (Oracle). Za razvoj se uporablja SVN. Pri nalaganju strani se program prebere iz baze podatkov. Najprej se procesira vsebina strani, ki se opremi s podatki iz podatkovne baze in/ali sistema SAP. Tako spremenjen XML dokument se procesira preko STYLESHEET-a, ki ga spremeni v HTML obliko, ter posreduje odjemalcu. V IML programih se združuje ukaze za prikaz podatkov, za spremembo in vpis ter oblikovanje strani. Za lažje razumevanje je priložena Slika 1.



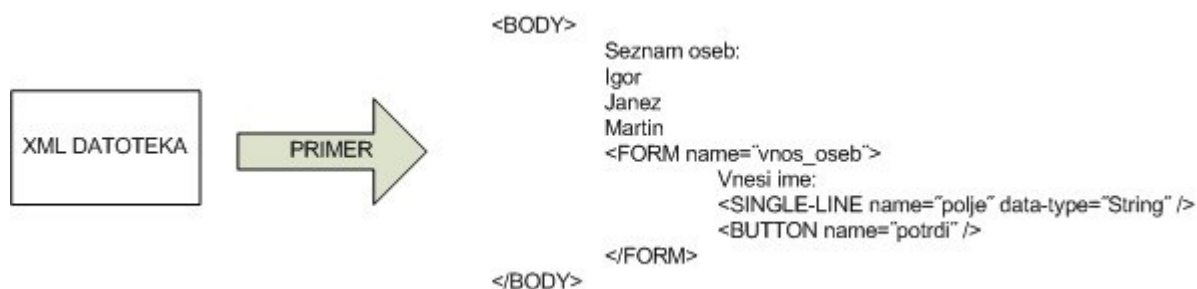
Slika 1: Postopek procesiranja IML programov.

Na sliki (Slika 2) je prikazan IML\_SOURCE. To je del kode, ki iz podatkovne baze pridobi seznam oseb, ter jih prikaže. Dodan pa je tudi obrazec, s katerim lahko vnesemo ime v podatkovno bazo.



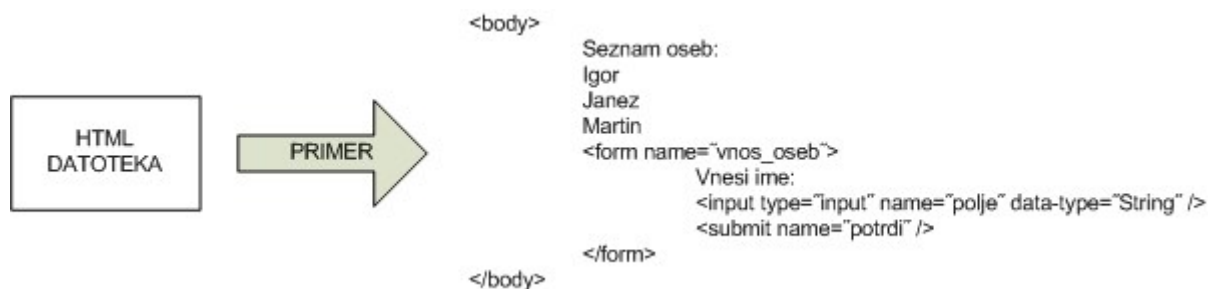
Slika 2: Primer IML programa.

Vsebina strani programa se ob prikazu procesira z JAVA SERVLET-om in tako se opremi s podatki iz podatkovne baze. Tako se ustvari nov XML dokument prikazan na sliki (Slika 3).



Slika 3: Sprocesiran XML dokument z JAVA SERVLET-om.

Nastali dokument se ponovno procesira z IML STYLESHEET-om in nastane končen HTML dokument (spletna aplikacija), kot je prikazano na sliki (Slika 4).



Slika 4: Končen rezultat procesiranja je HTML dokument.



Pisanje IML programov ima določena pravila, npr. oznake napisane z malimi črkami se pri procesiranju ne procesirajo, ostanejo nespremenjena. Oznake napisane z velikimi črkami pa se procesirajo preko IML STYLESHEET-a v HTML obliko.

Za delo z IML-jem je razvit dodaten plugin.

IML plugin v Eclipse omogoča naslednje:

- Vpis programov v podatkovno bazo.
- Lokalno izvajanje programov.
- Razhroščevanje lokalnih programov.
- Pridobitev opisa tabel iz Oracle v XML obliki
- Kreiranje in sprememba tabel v Oracle iz XML oblike

Posamezne programe ali skupino programov je možno tudi pognati preko razhroščevalnika za IML. V programu nastavimo eno ali več zaustavitvenih točk in ga poženemo. Ko bo izvajanje naletelo na to točko se bo ustavilo in nam prikazalo vrstico, v kateri se je program ustavil. Tako nam izpiše vsa imena in vrednosti spremenljivke, ki so v tej točki programa aktivne.

#### **4.4 Oracle SQL Developer**

Je brezplačno grafično orodje za razvoj podatkovnih baz. Z njim lahko pregledujemo podatkovne objekte, poganjamo SQL stavke in skripte ter urejamo in razhroščujemo PL/SQL stavke. SQL Developer povečuje produktivnost in poenostavlja zbirke podatkov razvojne naloge.[\[9\]](#)

## 5. Uporabljene metode

### 5.1 XML

XML (angl. Extensible Markup Language) je razširljiv označevalni jezik. Je preprost računalniški jezik podoben HTML-ju. Omogoča nam format za opisovanje strukturiranih podatkov in arhitekturo za prenos podatkov in njihovo izmenjavo med več omrežji.[\[11\]](#)

### 5.2 HTML

HTML (angl. Hyper Text Markup Language) je označevalni jezik namenjen izdelavi spletnih strani. Struktura in vsebina spletne strani sta predstavljeni z dokumentom, kateri vsebuje zaporedje ukazov, ki povedo kako naj se predstavljena stran prikaže. Ukazom rečemo z drugo besedo tudi značke, katere morajo biti vedno pravilno uporabljene (ugnezdene). Brskalnik s pomočjo pravilno strukturiranega dokumenta sestavi spletno stran. Vsak HTML dokument je sestavljen iz dveh glavnih delov. Prvi je glava (angl. HEAD) v katero se vpiše naslov strani (angl. TITLE), ki je vidna na naslovni vrstici brskalnika. Sledi mu telo (angl. BODY), v katerem je napisan celoten vidni del spletne strani.[\[4\]](#)

### 5.3 CSS

CSS (angl. Cascading Style Sheets) so kaskadne stilske podloge, ki omogočajo avtorjem spletnih strani, da predpišejo obliko posameznih elementov HTML dokumentov. Določamo lahko barve, velikosti, odmike, poravnave in še številne lastnosti elementov. Poleg tega lahko nadziramo aktivnosti, ki jih uporabnik nad elementi strani izvaja (npr. prekritje povezave z miško). CSS se je prvič pojavil leta 1996 in se do danes že veliko izpopolnil. Z novimi verzijami lahko nadziramo vse več elementov. Namen CSS-ja je ločitev strukture in vsebine spletne strani od njene oblikovne predstavitve. Tako omogočimo lažje urejanje stilov ter večjo preglednost HTML dokumentov. S CSS pravili se izognemo ponavljanju kode, saj omogočimo množici strani uporabo istih predlog. Poznamo dve skupini elementov (class in id), nad katerima določamo pravila posameznemu HTML elementu ter številnim sektorjem (<table>, <tr>, <p>, ...).[\[3\]](#)

## 5.4 JavaScript

JavaScript je marketinško ime za skriptni jezik ECMA Script in se uporablja za programiranje dinamičnih spletnih strani in aplikacij, ki se izvajajo na odjemalcu. Podpirajo ga vsi novejši spletni brskalniki. Veliko se uporablja v povezavi s knjižnico jQuery, katera programerjem olajša delo. JavaScript se v HTML vključi z namenom, da opravlja naloge, ki jih ni mogoče opravljati samo s statično stranjo. V programu JavaScript lahko napišemo funkcije, katerim pravimo skripte. Funkcije se izvedejo ob sprožitvi dogodka (npr. ob kliku na gumb, ob premiku miške,...).[\[1\]](#)[\[5\]](#)

## 5.5 SQL

SQL (angl. Structured Query Language) je najbolj razširjen in standardiziran povpraševalni jezik, za delo s podatkovnimi bazami. Omogoča nam kreiranje, spreminjanje, brisanje in manipulacijo s podatki shranjenimi v relacijski podatkovni bazi. Z nekoliko zahtevnejšo poizvedbo, kot je prikazana na sliki v prilogi [\[2\]](#) pa je mogoče pridobiti in izračunati marsikatero podatke.[\[8\]](#)

## 5.6 SVN

SVN ali s celotnim imenom Apache™ Subversion®, je sistem za verzioniranje dokumentov. Skrbi za shranjevanje posameznih programov, skupaj z informacijami o spremembi med katerimi je oseba, datum, komentar vpisa. Tako lahko vidimo celoten razvoj posameznega dokumenta. Vsebino dokumentov lahko primerjamo s starejšimi verzijami. Lahko tudi vračamo vsebino na starejše verzije, predvsem pa je pomembno, da so vsi dokumenti shranjeni na enem mestu.

Eclipse nam omogoča dokaj enostavno urejanje XML in XSL datotek, ter interakcijo s SVN repositorijem. SVN je z instalacijo Subversive - SVN Team integriran v Eclipse okolje. Za dostop do SVN-ja potrebujemo uporabniško ime in geslo.[\[10\]](#)

### 5.6.1 SVN – direktoriji

Strukturo shranjevanja dokumentov si moramo predstavljati podobno oz. enako kot strukturo direktorijev - map na disku. Posamezni direktoriji in programi v njih nosijo še dodatne informacije in shranjujejo posamezne verzije.

Po dogovoru je SVN organiziran v treh glavnih direktorijih:

- trunk - (deblo) mapa v kateri so shranjene tekoče verzije.
- branches - (veje) ko imamo v vidu večje spremembe, odpremo novo vejo. Tukaj dokumente dopolnjujemo in spreminjamo. Na koncu lahko vejo ponovno združimo v deblo, in postane tekoča verzija.
- tags - (oznake) v to mapo shranjujemo dokončne verzije, ki jih želimo shraniti, kot celoto.

### 5.6.2 SVN - merge urejanje konfliktov

Spremembe med posameznimi verzijami lahko pregledujemo in popravimo.



Slika 5: Pogled med dvema različnima verzijama programa.

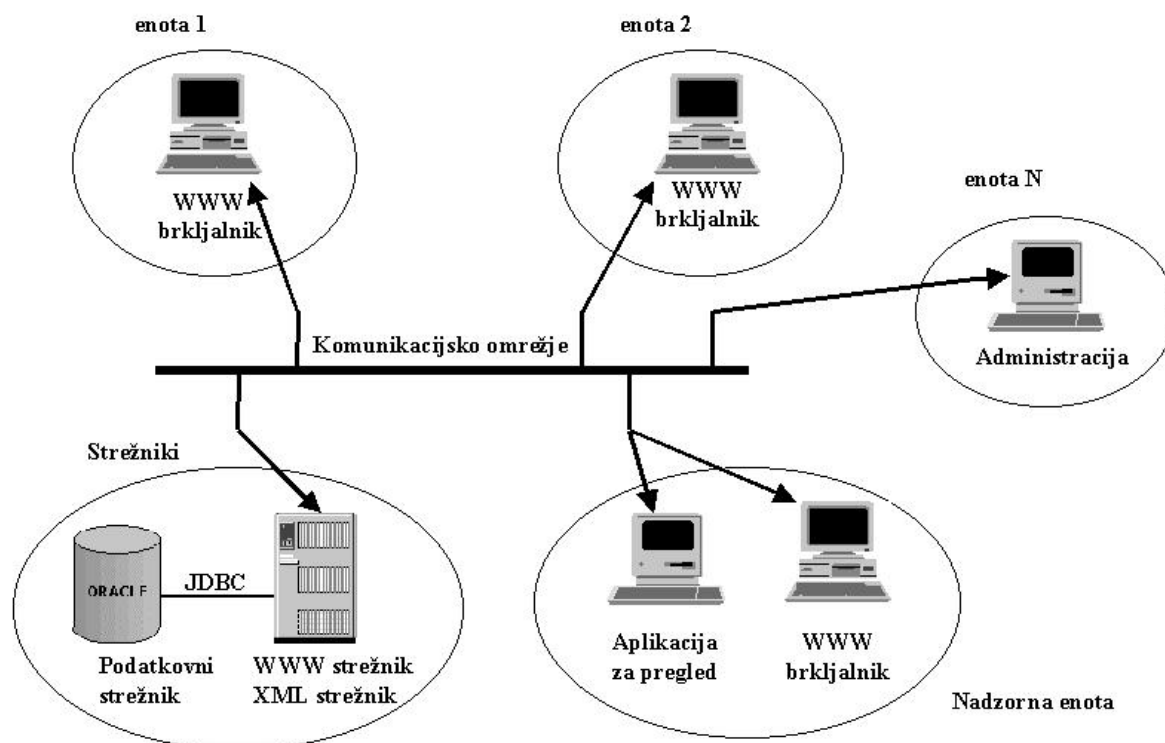
## 6. Razvoj aplikacije

Zahteve za razvoj aplikacije so bile znane že vnaprej, saj je bil namen aplikacije nadomestiti sedanji potek dela za FTQ sestanke. Zato je bila uporabljena metodologija razvoja, ki temelji na klasičnem zaporednem modelu. Razvoj aplikacije je potekal v podjetju, kjer je zaposlenih več programerjev. Zaradi varnosti je potekalo beleženje verzij aplikacije preko SVN-ja, tako se je tudi aplikacija shranila na strežnik, do katerega so lahko dostopali vsi uporabniki in programerji.

### 6.1 Arhitektura sistema

V podjetju je v uporabi že veliko število aplikacij, ki uporabljajo arhitekturo odjemalec-strežnik. Omenjena arhitektura je zelo pogosto uporabljena pri aplikacijah, do katerih dostopa večje število uporabnikov iz različnih lokacij, preko omrežja.

Način komunikacije je predstavljen na sliki (Slika 6). Odjemalec pošlje strežniku zahtevo za določeno aktivno stran, nato pa strežnik to izvrši. Na osnovi rezultata generira HTML spletno stran in jo pošlje nazaj odjemalcu. Pri tem ima ključni pomen omrežje, preko katerega poteka izmenjava podatkov in sporočil.



Slika 6: Prikaz arhitekture odjemalec strežnik, ter povezava med njima.

## 6.2 Podatkovna baza

V podjetju je že nameščena Oracleova podatkovna baza, katero uporabljajo že obstoječe aplikacije. Tudi ta nova aplikacija uporabljala že obstoječo podatkovno bazo, le da za hranjenje podatkov uporablja svoje tabele. Tabele in povezave med njimi so prikazane na sliki v prilogi [1]. Določeni podatki, ki jih aplikacija potrebuje za dodatne informacije so že shranjeni v drugih tabelah, zato ima dostop tudi do teh podatkov.

## 6.3 Uporabniški vmesnik

Pri načrtovanju uporabniškega vmesnika je bilo pomembno, da bo le ta čim bolj enostaven in preprost za uporabo.

Ključnega pomena pri razvoju uporabniškega vmesnika je imel že zgoraj opisani IML.

Uporabniki so do sedaj v elektronskih preglednicah hranili podatke o napakah na linijah kot prikazuje Slika 8. Uporaba nove aplikacije, pa ne bi smela veliko spremeniti načina beleženja, kajti drugače bi lahko imeli uporabniki pri vnosu napak težave.

Shranjeni podatki so tako zabeleženi v podatkovni bazi. Tam pa nam prav veliko ne pomenijo brez nekaterih vmesnikov, s katerimi bi lahko dostopali do njih. Zato so bile izdelane še nekatere funkcionalnosti aplikacije, ki nam omogočajo branje in izpisovanje teh podatkov.

Funkcionalnosti aplikacije za prikaz podatkov:

- Izpis napak za poljubno obdobje
- Mesečni pregled napak
- Tedenski izpis napak z grafoma

Izdelana pa je bila še ena funkcionalnost, ki je namenjena vzdrževanju šifrantov napak.

### 6.3.1 Vnos napak

Ta aplikacija je namenjena vnosu napak na posamezni liniji. Začrtana je tako, da poskuša na podoben način kot po starem (na papirju) zabeležiti napake, le da bi se te hranile v podatkovni bazi.

#### 6.3.1.1 Vnos napak na star način

Vnos napak je potekal na vsaki liniji ročno. Vsak dan je bilo potrebno pripraviti natisnjene preglednice z naborom vseh napak, ki se lahko pojavijo na liniji. Razlikovale so se po tipih izdelka, ki so ga izdelovali, to pomeni, da toliko tipov izdelkov kot so naredili v eni izmeni, toliko preglednic je bilo natisnjenih. Če je linija obratovala v jutranji in popoldanski izmeni je bilo potrebno pripraviti še dodatne preglednice, s katerimi so lahko ločili nastale napake po izmenah. Preglednica je prikazana na sliki (Slika 7). Namenjena je vsakodnevnemu vnosu napak za tip izdelka, ki je napisan na vrhu preglednice.

#### IZLOČENI KOSI NA LINIJI STATORJA BMW

OP S0550

Izločene statorje se izloči v rdeče zaboje za statorje, izločena ohišja oz. ohišja s statorjem pa v za to namenjen rdeč zaboj.  
! Obarvana količina predstavlja alarm in ob dosegu je potrebno obvestiti vodjo ODE oz. tehnologa.

DEL.NALOG:		TIP: BMW		DATUM:		IZMENA:	
------------	--	----------	--	--------	--	---------	--

NAPAKA	POSTAJA - VZROK	POJAVITEV																			
0-9	Splošna napaka - neznan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1001	S0100 - vstavljen že navit stator	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1002	S0100 - napačna višina statorja	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1003	S0100 - napačna višina statorja	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1100	S0150 - ni nalepke ali nečitljiva	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2020	S0220 - pretrgana žica	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
9900	S0550 - ročno odstranjen stator																				
	vzrok:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	vzrok:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	vzrok:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	Drugo:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	Drugo:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	Drugo:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

ŠTEVILO IZDELANIH STATORJEV	DOBRI:	SLABI:
-----------------------------	--------	--------

PREDDELAVEC:

Podpis: \_\_\_\_\_

Slika 7: Preglednica za vnos napak po starem.

Zabeležene napake so na koncu izmene prepisali v drugo preglednico, ki je bila podobna prvi, le da je bila prilagojena za cel mesec, kot prikazuje slika (Slika 8). Vpisala se je sešteta količina posamezne napake. Preglednice so se razlikovale po tipu izdelka in po izmeni v kateri so se napake pojavile.

Letrika						LISTA NAPAK														Linija: Montaža: Denis Koda : 11213127 Alfa										Oddelek : Mehatronika Mesec: September 2013					
Vrsta napake	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Slabo stator																																			
Polkodovani ležaj																																			
Slabo namagneten rotor																																			
Trenje																																			
Slabo vrtanjen ležaj																																			
Polkodovana termoizol. cevka																																			
Slabo priviti PLP																																			
Hrupen motor																																			
Slaba sestava reža																																			
Nepravilno vstavljen viskočnik																																			
Ke																																			
5th harmonic																																			
7th harmonic																																			
Žica izven mase																																			
Skupaj napak																																			
Izdelanih																																			

Slika 8: Elektronska preglednica za hranjenje napak.

V obe preglednici, ki so prikazane na sliki so podatki zapisani ročno. Iz preglednice, ki je na sliki (Slika 8), so se v začetku vsakega tedna prepisale napake, ki so se pojavile prejšnji teden v elektronsko preglednico, kjer so se dokončno hranile.

Iz teh elektronskih preglednic se ja nato seštelo istovrstne napake prejšnjega tedna iz vseh tipov izdelka, ter izpisalo pet napak, ki so se najpogosteje pojavile, ter izrisalo grafe, ki so prikazani na sliki (Slika 16).

Skratka, če je linija izdelovala v enem dnevu dva različna tipa izdelka v jutranji in popoldanski izmeni je bilo treba za en dan pripraviti štiri različne preglednice. Še večji problem so predstavljale shranjene elektronske preglednice, saj je bilo potrebno za tako linijo vsak mesec shraniti kar nekaj različnih elektronskih preglednic. Primer če se je izdelovalo osem različnih tipov izdelka v dveh izmenah je v enem mesecu nastalo 16 elektronskih preglednic. Hraniti so se morale za najmanj tri leta nazaj. Podjetje ima zelo veliko število liniji, kar pomeni zelo veliko število elektronskih preglednic. Poleg tega je še pogled nad shranjenimi napakami zelo slab in s tem je tudi težko izvajati kakršne koli analize.



### 6.3.1.2 Vnos napak z aplikacijo

Uporaba aplikacije za vnos napak omogoča veliko lažji zajem in beleženje napak, ki se pojavijo na posamezni liniji. Namenjena je za vse linije v podjetju, zato je tudi temu primerno prilagojena.

Aplikacija je dostopna na intranetu, ki je v lasti podjetja. Do nje lahko dostopajo uporabniki, ki imajo dodeljeno vlogo za dostop do aplikacije. Za preverjanje vloge je na začetku programa naslednja koda (Slika 9). V tem primeru je za dostop do aplikacije potrebno imeti vlogo IKO\_UP\_FTQ.

```
<IML title="FTQ prikaz" layout="/IMLDefault.xml" protect="R:IKO_UP_FTQ" content-type="{CT}" version="1.0">
```

Slika 9: Izsek kode za preverjanje vloge.

Ob prijavi uporabnika na intranet je iz prijave razvidno na katero linijo je uporabnik nameščen. Zato aplikacija že ob zagonu prepozna uporabnika in s tem mu omeji pogled nad delovnimi nalogami, tako da mu prikaže samo tiste, ki so napisani za tisto linijo kjer dela.

Aplikacija uporabniku ob vstopu ponudi vnosno polje, kje mora vnesti številko delovnega naloga na katerem se izdelujejo izdelki. Včasih se zgodi, da se delovni nalog predčasno zavrže ali izgubi. Aplikacija je izdelana tako, da nam omogoča vpogled nad odprtimi delovnimi nalogami. Za taki vpogled je potrebna le ena vrstica programske kode, katera pa uporablja za pridobitev podatkov zelo zahtevno poizvedbo, kot je prikazano v prilogi [3].

### Vnos napak

Operacija/Delovni nalog:

Lista vrednosti - Mozilla Firefox

intranet.group.letrika.com/intra/iml/ipro/ipro/\$\$LOV.HTM


Išči:

I	Obrat	Lansiran	Operacija	Nalog	Koda	Naziv
X	0045	28.02	0001	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0005	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0100	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0102	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0155	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0158	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0160	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0170	798471	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0155	818305	16282477	stator AME64
X	0045	28.02	0160	818305	16282477	stator AME64
X	0045	28.02	0200	818305	16282477	stator AME64
X	0045	28.02	0005	818316	16350231	stator z navitjem AME64
X	0045	28.02	0001	818593	11213239568	motor 12V 0,3kW AME6423
X	0045	28.02	0005	818593	11213239568	motor 12V 0,3kW AME6423

Slika 10: Okno za izbor odprtih delovnih nalogov.

Ob izboru delovnega naloga nam samodejno najde za katero linijo gre in kateri izdelek se na liniji izdeluje trenutno. Ponudi nam tudi izbor osnovne delovne skupine (ODS).


## Vnos napak


Operacija/Delovni nalog:	0017361697	...
Datum:	03.03.14	
Delovni nalog:	818305	
Linija:	Ročno DM, SM 1115005 (zalivanje. DELPHI)	
Številka izdelka:	16282477	
Naziv izdelka:	stator AME64	
<b>ODS</b>		
MEH-ODS 2-3	Potting 1	
MEH-ODS 2-4	Potting 2	
MEH-ODS 2-5	Potting 3	
MEH-ODS 2-6	Bužirke	

Slika 11: Okno za izbor osnovne delovne skupine (ODS).

S klikom na enega od naštetih se nam pojavi seznam napak in možnost vnosa količine posamezne napake, ter število dobrih izdelkov, ki so jih izdelali. Pred začetkom vnosa moramo biti pozorni na datum, običajno je nastavljen na dan za katerega vnašamo, imamo pa tudi možnost, da datum spremenimo. To nam omogoča vnos napak za nazaj.

## Vnos napak

Operacija/Delovni nalog:  

Datum:  

Delovni nalog: **818305**

Linija: **Ročno DM, SM 1115005 (zalivanje. DELPHI)**

Številka izdelka: **16282477**

Naziv izdelka: **stator AME64**

ODS	
MEH-ODS 2-3	Potting 1
MEH-ODS 2-4	Potting 2
MEH-ODS 2-5	Potting 3
MEH-ODS 2-6	Bužirke

Napake	Vnos količine	Količina
Število dobrih	<input type="text"/>	
Skupaj že vnešene napake		
BIT	<input type="text"/>	
Ekscentrično zalit stator	<input type="text"/>	
Mehanske poškodbe	<input type="text"/>	
Nepravilno vstavljen odcep v zalivni pokrov	<input type="text"/>	
Ostanki mase v ujemu okrova	<input type="text"/>	
Ostanki tampona v statorju	<input type="text"/>	
Prelit stator	<input type="text"/>	
Premalo zalit	<input type="text"/>	
Pušča lonec	<input type="text"/>	
Slab stator	<input type="text"/>	
Umazan stator	<input type="text"/>	
Utrgana žica	<input type="text"/>	
Zvito uho	<input type="text"/>	
Žica v območju rotorja	<input type="text"/>	

Slika 12: Obrazec za vnos napak.

Uporabnik tako na koncu izmene oz. ko prične izdelovati drugi tip izdelka vpiše nastale napake in število dobrih izdelanih izdelkov, ter pritisne gumb »Vnos«. Ob vnosu aplikacija shranjuje podatke v podatkovno bazo v dve tabeli in sicer v prvo tabelo (IPRO\_RALIZACIJA) shrani število izdelanih izdelkov, datum in podatke o liniji. V drugo (IPRO\_NEKAKOVOSTI\_POROC) pa količino posamezne napake in tuj ključ, ki je primarni ključ v prvi tabeli. S tem sta obe tabeli povezani.

## 6.3.2 Izpis napak

Aplikacija ima funkcijo za izpis napak, s katero lahko vidimo napake, ki so bile vpisane z aplikacijo za vnos le teh. Omogoča nam številne možnosti izpisa. Dostop do nje je ravno tako omejen, saj se je za zagon potrebno prej prijaviti na intranet. Namenjena je vodjem proizvodnje za vpogled nad napakami in izvajanjem različnih analiz.

Ob zagonu se pojavijo vse napake, ki so bile na linijah vpisane prejšnji teden in za katere je zadolžen vodja. Na vrhu je meni s katerim lahko filtriramo napake (Slika 13). Možnosti za filtriranje napak so:

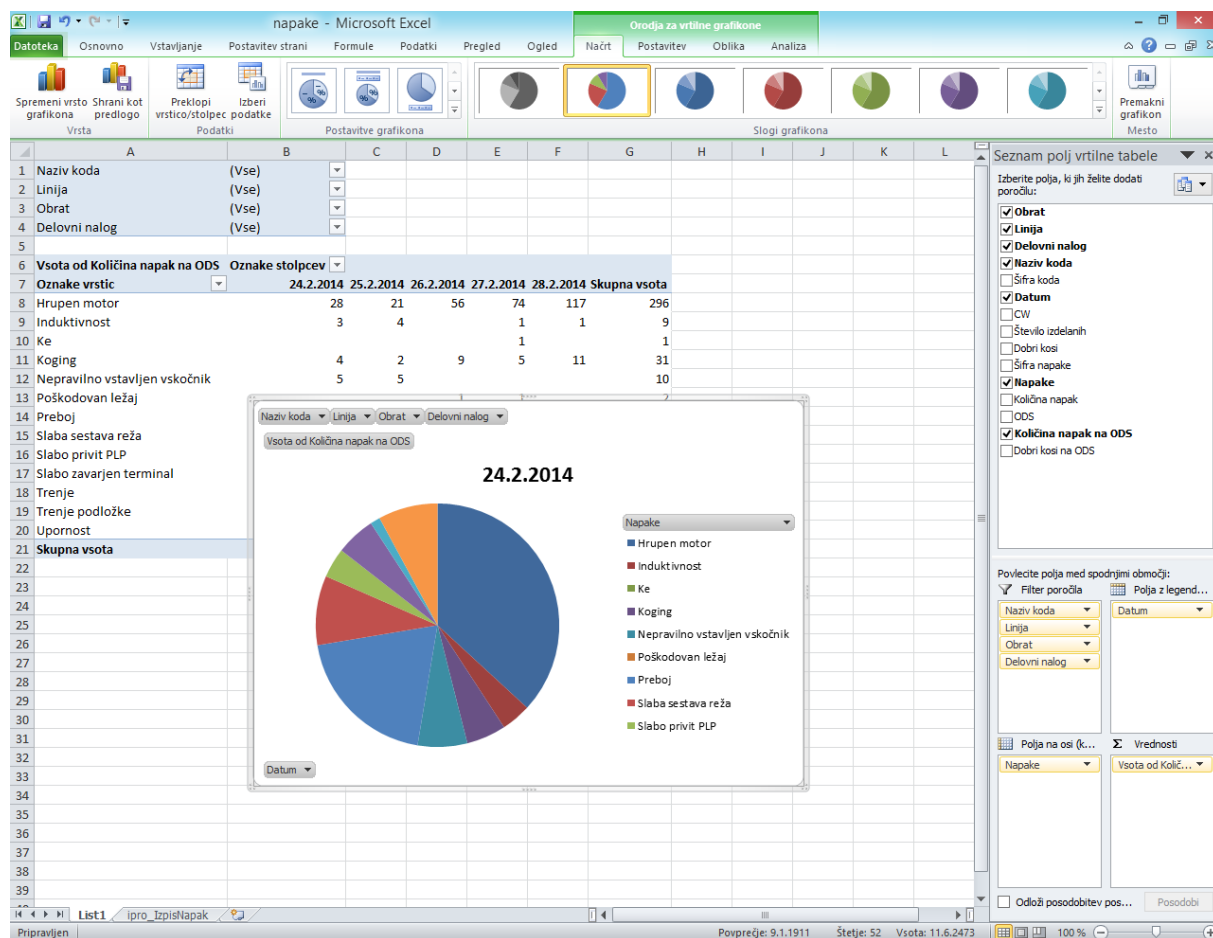
- Izbor obdobja za prikaz (običajno je nastavljeno na prejšnji teden)
- Izbor napak glede na obrat
- Izbor napak glede na linijo
- Izbor napak glede na posamezen ODS na liniji

### Izpis napak

Datum od 24.02.2014		do 02.03.2014		Obrat 0045		Linija 1115003 OPE-2 montaža EPS		ODS -						
Obrat	Linija	Delovni nalog	Naziv koda	Šifra koda	Datum	CW	Število izdelanih	Dobri kosi	Šifra napake	Napake	Količina napak	ODS	Količina napak na ODS	Dobri kosi na ODS
0045	Montažna linija EPS DELPHI	798465	motor 12V 0,25kW AME6401	11213105568	24.02.14	09	2046	1970		Hrupen motor	28 MEH-ODS 2-1		8	1000
											MEH-ODS 2-2		20	970
										Induktivnost	3 MEH-ODS 2-2		3	970
										Koging	4 MEH-ODS 2-2		4	970
										Nepravilno vstavljen vskočnik	5 MEH-ODS 2-1		5	1000
										Preboj	15 MEH-ODS 2-1		13	1000
											MEH-ODS 2-2		2	970
										Slaba sestava reža	7 MEH-ODS 2-1		1	1000
											MEH-ODS 2-2		6	970
										Slabo privit PLP	3 MEH-ODS 2-1		3	1000
										Slabo zavarjen terminal	4 MEH-ODS 2-1		4	1000
										Trenje	1 MEH-ODS 2-2		1	970
										Trenje podložke	6 MEH-ODS 2-1		1	1000
											MEH-ODS 2-2		5	970

Slika 13: Izpis napak z določenimi filtri.

Izpisane napake lahko prenesemo v elektronsko preglednico, kjer lahko izvajamo analize nad napakami npr. risanje raznih grafov (prikazuje Slika 14), računanje odstotkov napak v nekem obdobju, ...



Slika 14: Prikaz napak na grafu v odstotkih.

### 6.3.3 Mesečni izpis napak

Ta funkcija je podobna funkciji za izpis napak, le da ima nekaj manjših razlik. Za dostop do nje potrebujemo ravno tako kot pri ostalih aplikacijah prijavo. Napake lahko filtriramo po linijah, ODS-ih in mesecih. Napake so prikazane za obdobje enega meseca. Prikaz je tabelaričen, kot prikazuje slika (Slika 15). Napake, ki se pojavijo v enem dnevu so seštete iz vseh izmen, takrat ko je linija obratovala. Na koncu je še seštevek vseh napak po dnevih in količina izdelkov brez napak. Skrajno desno pa seštevek posamezne napake v obdobju enega meseca.

### Mesečni pregled napak

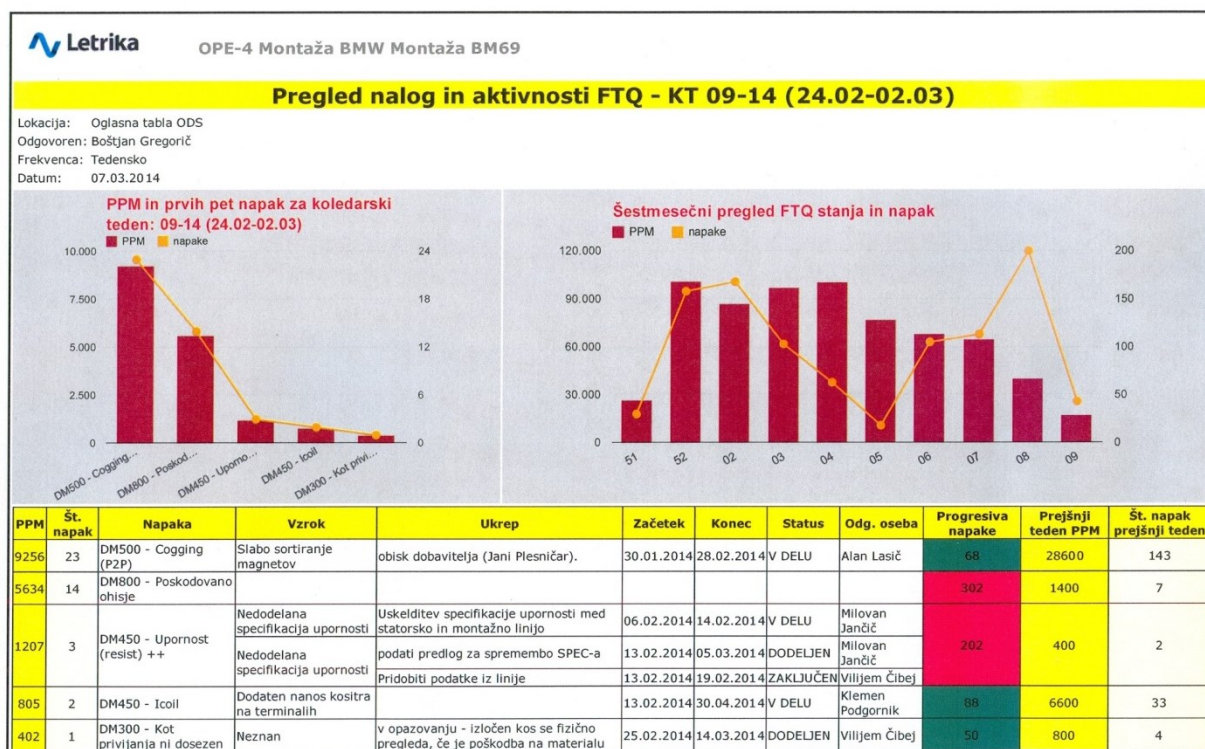
Linija:  ODS:  03-14 (01.03.14-31.03.14)

Napaka	09/14		10/14							13/14							14/14	Skupaj
	01	02	03	04	05	06	07	08	09	24	25	26	27	28	29	30	31	
DM200 - NOK																		
DM300 - Kot privijanja ni dosežen													3					10
DM300 - Kot privijanja prekoračen																		2
DM300 - Moment privijanja ni dosežen																		
DM300 - Navor																		
DM450 - 5TH Harmonik																		
DM450 - 7TH Harmonik																		
DM450 - Icoil												3		1				4
DM450 - Induktivnost																		1
DM450 - Ke																		1
DM450 - Upornost (Resist) --					8							1		5				16
DM450 - Upornost (resist) ++																		4
DM450 - Vpetje												3						5
DM500 - Cogging (P2P)					14	17	7			21	80	57	75	35			2	562
DM550 - Trenje													1					1
DM800 - Kontaminacija																		
DM800 - Neodstranjena nalepka																		
DM800 - Nepravilna sestava																		
DM800 - Poškodovan ZLP																		
DM800 - Poškodovan konektor																		3
DM800 - Poškodovan ležaj																		
DM800 - Poškodovan neprivit vijak																		
DM800 - Poškodovana gred rotorja																		
DM800 - Poškodovani zatiči																		
DM800 - Poškodovano ohišje				6	14	2	3				3	2	5	7			4	78
DM800 - Ujeto tesnilo																	3	18
Reklamacija kupca																		
Skupaj napak:				6	36	19	10			21	86	63	84	48			14	712
Skupaj dobrih kosov:				568	1216	257	720			385	812	653	638	600			639	12859

Slika 15: Mesečni izpis napak.

### 6.3.4 Tedenski izpis napak z dvema grafoma

Zakonodaja predpisuje, da mora imeti vsaka linija na listu papirja natisnjenih pet napak, ki so se pojavile prejšnji teden v največjem številu ter dva grafa. Poleg napak pa morajo biti napisani še vzroki za nastanek, ter ukrep s katerim naj bi bila napaka odpravljena. Prvi graf prikazuje pet napak v odvisnosti s PPM-om (angl. percent per milion), drugi pa po tednih številu pojavljenih napak v odvisnosti s PPM-om za pol leta nazaj.



Slika 16: Pripravljen izpis napak z grafoma za oglasno tablo.

Do sedaj so ta list pripravili ročno v elektronski preglednici. Da bi se znebili zamudi za pripravo takega dokumenta, aplikacija vsebuje funkcijo za izdelavo dokumenta. Prav tako kot pri vseh do sedaj opisanih funkcijah, se je za dostop do nje potrebno prijaviti preko intraneta. Ob zagonu se nam pojavi meni za izbiro linije. Ob izboru linije se prikaže še en meni v katerem lahko izberemo ODS, polje z datumom po tednih in dva gumba. Razlika med njima je samo ta, da eden omeji izpis na pet napak, drugi pa prikaže vse napake, ki so se pojavile na liniji za izbrani teden. Slika 17 prikazuje možnost filtriranja po vrsti linije in osnovni delovni skupini za izpis, ki je prilagojen za oglasno tablo.

## FTQ izbor

Linija:  ODS:  10-14 (03.03-09.03)

Slika 17: Izbor linije in osnovne delovne skupine za prikaz.

Ob pritisku na gumb, se nam pojavi novo okno, v katerem so prikazani podatki v popolno enaki obliki kot v dokumentih, ki so jih uporabljali do sedaj. Uporabnik natisne prikazane podatke in tako je že pripravljen list z napakami ter grafi.

### 6.3.5 Šifrant napak

Vse do sedaj opisane funkcije aplikacije uporabljajo seznam napak, ki so shranjene v podatkovni bazi. Za dostop do teh napak je bila izdelana funkcija za vzdrževanje šifranta napak. Namenjena je shranjevanju, urejanju in brisanju napak za posamezno linijo. Ob zagonu se nam pojavi seznam liniji, za katere je oseba, ki uporablja aplikacijo zadolžena. S klikom na eno izmed prikazanih se odpre seznam napak, ki so pripisane izbrani liniji. O kliku na napako se nam podatki o napaki prepišejo v okno za urejanje. Izbrano napako lahko uredimo, dodamo (a z drugim imenom) ali pa izbrišemo. V polju za urejanje imamo štiri vnosna polja, kjer lahko vpišemo šifro in naziv napake. Polje vpliv na K2 in alarm se za enkrat še ne uporabljata za nobeno funkcionalnost. Namen polja vpliv na K2 pa je ta, da pri posamezni napaki povemo, kolikšna je za nastanek napake krivda delavca. To naj bi se nato uporabljalo pri računanju stimulacije delavcev na posamezni liniji. Polje alarm pa naj bi bila količina napak, pri kateri naj bi se pojavilo opozorilo in pri tem se bi tudi obvestilo določene osebe o prekoračitvi alarma.



## Vzdrževanje šifranta napak

LINJA		
1115002	OPE-4 Stator ZF	Ročno DM na SM 1115002
1115003	OPE-2 montaža EPS	Montažna linija EPS DELPHI
1115004	OPE-3 stator	Stroj za vst. izolac. stat. DELPHI EPS
1115005	OPE-2 poting EPS	Ročno DM, SM 1115005 (zalivanje. DELPHI)
1115006	OPE-4 Stator Comstar	Ročno DM na SM 1115006
1115006	OPE-4 Stator Comstar	Varilna naprava
1115007	OPE-1 stator Busch - Poting	Splošno DM na STM 1115007
1115008	OPE-1 Montaža Busch	Montaža in magnetenje
1115009	OPE-4 Stator BMW	Linija statorja ATS (BM69)
<b>1115010</b>	<b>OPE-4 Montaža BMW</b>	<b>Montaža BM69</b>
1115011	Stator Busch - Saleri - navijanje	Naprava za sest. zolacijskih kletk

Šifra napake	Napake	Vpliv na K2 %	Alarm
	DM550 - Trenje		
	DM800 - Kontaminacija		
	DM800 - Neodstranjena nalepka		
	DM800 - Nepravilna sestava		
	DM800 - Poškodovan ZLP		
	DM800 - Poškodovan konektor		
	DM800 - Poškodovan ležaj		
	<b>DM800 - Poškodovan neprivit vijak</b>		
	DM800 - Poškodovana gred rotorja		
	DM800 - Poškodovani zatiči		
	DM800 - Poškodovano ohišje		
	DM800 - Ujeto tesnilo		

Vnos/sprememba napake	
Šifra napake	<input type="text"/>
Naziv napake	<input type="text" value="DM800 - Poškodovan neprivit vijak"/>
Vpliv na K2	<input type="text" value=""/> %
Alarm	<input type="text"/>

Slika 18: Del aplikacije za vzdrževanje šifranta napak.

## 7. Sklepne ugotovitve

Spletna aplikacija za vnos in izpis napak bo zaposlenim v podjetju poenostavila in olajšala delo. Ključnega pomena so podatki shranjeni v podatkovni bazi in ne več elektronske preglednice shranjene na podatkovnih diskih. Tako je dostop do podatkov veliko lažji in hitrejši.

Aplikacija je izdelana tako, da omogoči vse dogovorjene zahteve in zastavljene cilje, kateri so bili zadani na začetku. Zaključek diplomskega dela je delujoča spletna aplikacija, katera se že uporablja v nekaterih delih podjetja.

Uporabniki so z aplikacijo zelo zadovoljni, saj imajo zelo učinkovit pogled nad napakami in s tem lahko zelo hitro ukrepajo, da se napake hitreje odpravi.

S selitvijo aplikacije na preostale dele podjetja, bodo v prihodnje potrebne še nekatere nadgradnje in izboljšave. Nadgradnja je mišljena predvsem pri delu aplikacije za vnos napak. Tam bi bilo možno vnašati napake s čitalcem črtnih kod direktno v podatkovno bazo, brez da bi predhodno beležili napake na papir. S tem bi še dodatno prihranili papir in čas, ki je potreben za prepis podatkov iz papirja v aplikacijo. Pri šifrantu napak ima aplikacija možnost vnosa alarma in vpliv na K2. Atribut alarm bo ob nadgradnji aplikacije, ko bo le ta beležila napake s čitalcem črtnih kod namenjeno temu, da ob prekoračitvi količine napak aplikacija samodejno opozori določene osebe o prekoračitvi alarma za posamezno napako. Atribut vpliv na K2 pa je bilo dodano na željo nekaterih zaposlenih. Namenjeno pa bo za računanje stimulacije zaposlenih na posamezni liniji.

Aplikacijo naj bi v prihodnosti začeli uporabljati tudi nekateri dobavitelji.

Zabeleženi podatki so uporabljeni tudi z drugo aplikacijo, ki je namenjena vodenju sestankov FTQ. Tam se napakam dodelijo vzroki in ukrepi, s katerimi naj bi se jih odpravilo.

## Literatura

- [1] Mark E. Daggett, *Expert JavaScript*, 2013, str. 131-150.
- [2] Apache Tomcat Server. Dostopno na:  
[http://en.wikipedia.org/wiki/Apache\\_Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat)  
<http://tomcat.apache.org/>
- [3] Stilne predloge CSS. Dostopno na:  
<http://sl.wikipedia.org/wiki/CSS>  
[http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [4] Označevalni jezik HTML. Dostopno na:  
<http://en.wikipedia.org/wiki/HTML>  
<http://sl.wikipedia.org/wiki/HTML>
- [5] Skriptni programski jezik JavaScript. Dostopno na:  
<http://sl.wikipedia.org/wiki/JavaScript>  
<http://en.wikipedia.org/wiki/JavaScript>
- [6] Predstavitev podjetja. Dostopno na:  
<http://www.letrika.com/sl/skupina-letrika/o-skupini/predstavitev/>
- [7] Razvojno okolje Eclipse. Dostopno na:  
[http://en.wikipedia.org/wiki/Eclipse\\_%28software%29](http://en.wikipedia.org/wiki/Eclipse_%28software%29)
- [8] Strukturirani povpraševalni jezik SQL. Dostopno na:  
<http://en.wikipedia.org/wiki/SQL>  
<http://sl.wikipedia.org/wiki/SQL>
- [9] Razvojno orodje SQL Developer. Dostopno na:  
<http://www.oracle.com/technetwork/developer-tools/sql-developer/what-is-sqldev-093866.html>
- [10] Sistem za verzioniranje SVN. Dostopno na:  
[http://en.wikipedia.org/wiki/Apache\\_Subversion](http://en.wikipedia.org/wiki/Apache_Subversion)

[11] Označevalni jezik XML. Dostopno na:

<http://sl.wikipedia.org/wiki/XML>

<http://en.wikipedia.org/wiki/XML>

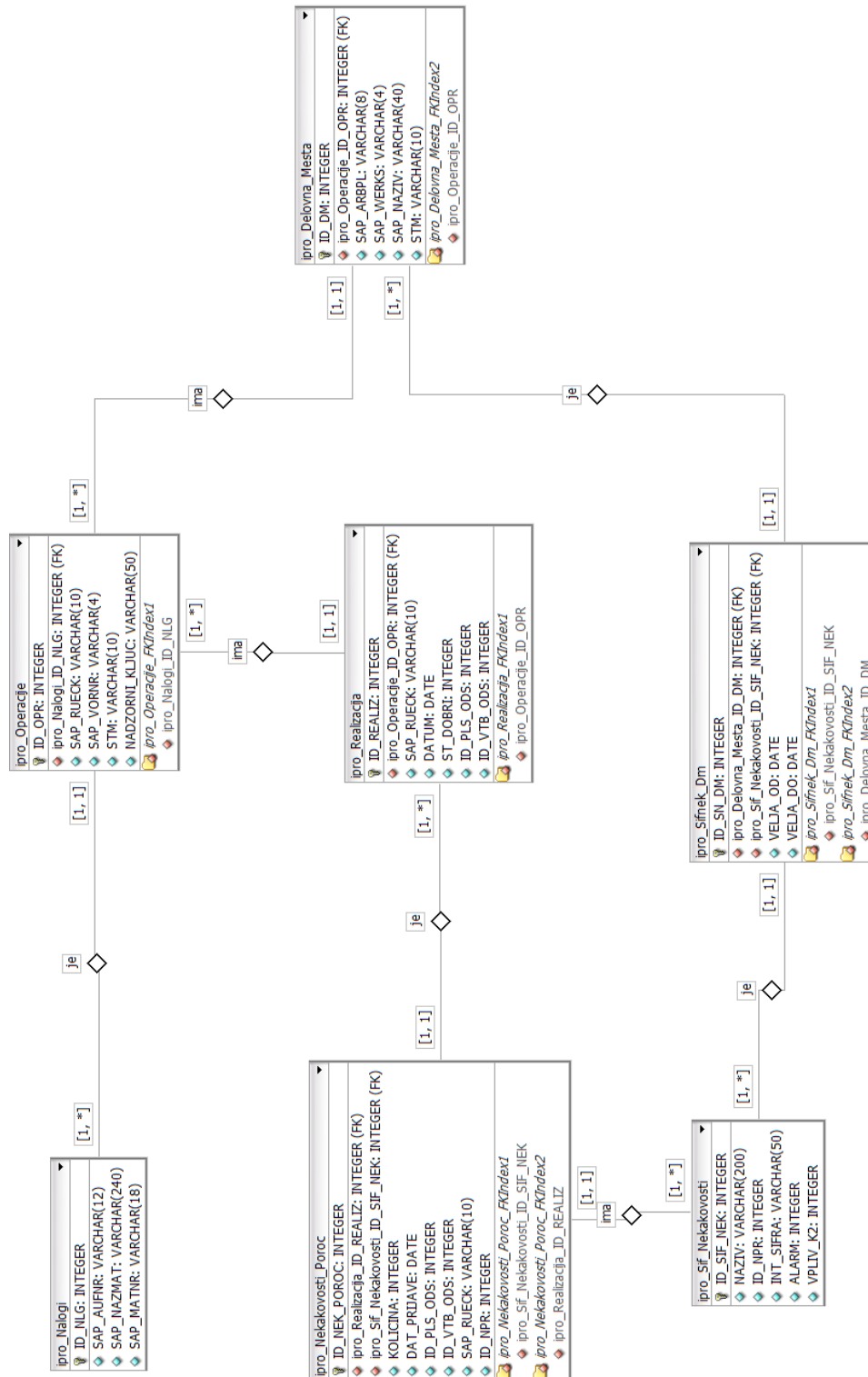
## Seznam slik

Slika 1: Postopek procesiranja IML programov.....	5
Slika 2: Primer IML programa. ....	6
Slika 3: Sprocesiran XML dokument z JAVA SERVLET-om. ....	6
Slika 4: Končen rezultat procesiranja je HTML dokument.....	6
Slika 5: Pogled med dvema različnima verzijama programa. ....	10
Slika 6: Prikaz arhitekture odjemalec strežnik, ter povezava med njima. ....	11
Slika 7: Preglednica za vnos napak po starem.....	13
Slika 8: Elektronska preglednica za hranjenje napak. ....	14
Slika 9: Izsek kode za preverjanje vloge. ....	15
Slika 10: Okno za izbor odprtih delovnih nalogov.....	15
Slika 11: Okno za izbor osnovne delovne skupine (ODS). ....	16
Slika 12: Obrazec za vnos napak. ....	17
Slika 13: Izpis napak z določenimi filtri.....	18
Slika 14: Prikaz napak na grafu v odstotkih. ....	19
Slika 15: Mesečni izpis napak. ....	20
Slika 16: Pripravljen izpis napak z grafoma za oglasno tablo. ....	21
Slika 17: Izbor linije in osnovne delovne skupine za prikaz. ....	22
Slika 18: Del aplikacije za vzdrževanje šifranta napak. ....	23

# Priloge

## Priloga 1

Konceptualni model podatkovne baze.



## Priloga 2

Poizvedba, katera izpiše vse možne ODS-e za izbrano linijo.

```
<SELECT name="BERI_ODS" nuls="yes">
with dat_stm as (
    select dn.dat_lans
        , o.stm
    from IPRO.IPRO_OPERACIJE o
    join IPRO.IPRO_NALOGI DN ON DN.ID_NLG = o.id_nlg
    where o.SAP_RUECK = {DELOVNI_NALOG}
)
, moja_oe as (
    select d.id_pls_oge as id_pls
        , d.id_vtb_oge as id_vtb
    from atr_uporabniki_vloge u
    join kdr_osebe o on u.id_osb = o.id_pls and u.id_vtb = o.id_vtb
    join kdr_odlocbe_za_dlm d on d.id_osb = o.id_osb and
        trunc(sysdate) between d.datum_od and utl_date.nvlmax(d.datum_do)
    where u.id_upb = {$USER_ID} and u.tip = 'U'
)
, odsi as (
    select e.*
    from table (im_user.role_getobjects({$USER_ID},
        (select id_upb
            from atr_uporabniki_vloge
            where kr_naziv='IKO_VN_NAP' and tip='V'),
        'OE-ON-PATH')) h
    join pls_pos_sub_pt e on h.object_id = e.id_pls and 1 = e.id_vtb and
        e.sif_2 = 'ODS'
)
, nase_sm as (
    select k.id_pls
        , k.id_vtb
        , dat_stm.dat_lans
    from kdr_pls_pos_sub k
    join dat_stm on k.sif_1 = dat_stm.stm and dat_stm.dat_lans between
        k.velja_od and k.velja_do
)
, hiar as (
    select h.id_pls
        , h.id_vtb
        , h.id_pls_nad
        , h.id_vtb_nad
        , h.z_st
    from pls_ia_raz_hiar h
    join dat_stm on dat_stm.dat_lans between h.velja_od and h.velja_do
)
select odsi.SIF_1 as ODS_VALUE
    , odsi.NAZ_3 as ODS_NAZIV
    , odsi.id_pls as ods_id_pls
    , odsi.id_vtb as ods_id_vtb
    , nase_sm.dat_lans
    , moja_oe.id_pls as moj_id_pls
    , count(*) over () as st_ods
    , re.id_realiz as id_realiz1
from odsi
```

```

        join hiar on odsi.id_pls = hiar.id_pls_nad and odsi.id_vtb = hiar.id_vtb_nad
        JOIN NASE_SM ON HIAR.ID_PLS = NASE_SM.ID_PLS AND HIAR.ID_VTB = NASE_SM.ID_VTB
left join moja_oe on moja_oe.id_pls = odsi.id_pls and moja_oe.id_vtb =
        odsi.id_vtb
left join IPRO.IPRO_REALIZACIJA re on re.del_mesto = {DEL_MEST01}
        and re.obrat = {OBRAT1}
        and re.sap_rueck = {DELOVNI_NALOG}
        and re.ID_PLS_ODS = odsi.id_pls
        <SWITCH><CASE>and re.DATUM = {DATUM}</CASE>
        <CASE>and re.DATUM = trunc(sysdate)</CASE>
        </SWITCH>
order by nvl2(moja_oe.id_pls,1,2), odsi.sif_1
</SELECT>

```



## Priloga 3

Vrstica programske kode, katera prikaže vnosno polje in poleg njega še gumb za iskanje.

```
<td colspan="3">  
  <SINGLE-LINE name="VNOS" size="12" onchange="pritisnjen();" onfocus="this.select();" lov="LOV_NAJDI_DN"/>  
</td>
```

Ob pritisku na gumb se v ozadju izvede naslednja poizvedba, katera prikaže vse odprte delovne naloge, za katere je uporabnik avtoriziran.

```
<LOV name="LOV_NAJDI_DN" return="SAP_RUECK:VNOS" hidden="SAP_RUECK">  
  with dat_stm as (  
    select dn.dat_lans  
      , o.STM  
      , o.SAP_RUECK  
      , dn.sap_aufnr  
      , dn.sap_matnr  
      , dn.sap_nazmat  
      , o.obrat  
      , o.del_mesto  
      , o.sap_vornr  
    from IPRO.IPRO_OPERACIJE o  
    join IPRO.IPRO_NALOGI DN ON DN.ID_NLG = o.id_nlg  
    where dat_lans between trunc(sysdate) - 7 and trunc(sysdate) + 1 and  
      (o.obrat , o.del_mesto) in (select SAP_WERKS  
                                , SAP_ARBPL  
                                from IPRO.IPRO_DELOVNA_MESTA)  
  )  
  , auth as (select distinct object_id  
    from table (im_user.role_getobjects({$USER_ID},  
      (select id_upb  
        from atr_uporabniki_vloge  
        where kr_naziv='IKO_VN_NAP' and  
          tip='V'), 'OE-ON-PATH'))  
  )  
  , nase_sm as (select k.id_pls  
    , k.id_vtb  
    , dat_stm.DAT_LANS  
    , dat_stm.SAP_RUECK  
    , dat_stm.sap_aufnr  
    , dat_stm.sap_matnr  
    , dat_stm.sap_nazmat  
    , dat_stm.obrat  
    , dat_stm.del_mesto  
    , dat_stm.sap_vornr  
    from kdr_pls_pos_sub k  
    join dat_stm on k.sif_1 = dat_stm.stm and  
      dat_stm.dat_lans between k.velja_od and k.velja_do  
  )  
  , hiar as (select h.id_pls  
    , h.id_vtb  
    , h.id_pls_nad  
    , h.id_vtb_nad  
    from pls_ia_raz_hiar h  
    where trunc(sysdate) + 1 >= h.velja_od and
```

```

        trunc(sysdate) - 7 <= h.velja_do
    )
, odsi as (SELECT hiar.id_pls as id_pls_stm
               , hiar.id_vtb as id_vtb_stm
               , e.velja_od
               , e.velja_do
            from auth a
            join kdr_pls_pos_sub e on a.object_id = e.id_pls and
                1 = e.id_vtb and e.sif_2 = 'ODS'
            join hiar on hiar.id_pls_nad = e.id_pls and
                hiar.id_vtb_nad = e.id_vtb
            where trunc(sysdate) + 1 >= e.velja_od and
                trunc(sysdate) - 7 <= e.velja_do
        )
select nase_sm.obrat as obrat
      , to_char(nase_sm.dat_lans,'DD.MM') as lansiran
      , nase_sm.sap_vornr as operacija
      , ltrim(nase_sm.sap_aufnr,'0') as nalog
      , nase_sm.sap_matnr as koda
      , nase_sm.sap_nazmat as naziv
      , nase_sm.SAP_RUECK as SAP_RUECK
from NASE_SM
where exists (select null from odsi where ODSI.ID_PLS_STM = NASE_SM.ID_PLS
              AND ODSI.ID_VTB_STM = NASE_SM.ID_VTB and
              nase_sm.dat_lans between odsi.velja_od and odsi.velja_do)
order by obrat, lansiran desc, nalog, operacija
</LOV>

```